Real-time Synthesize Video Streaming to Another based on 3D Gaussian Splatting: A Proposal

Siyan HU, # 12217679

I. NeRF

NeRF represents the future of the research in 3D scene reconstruction, and has already been transferred into multiple applications, such as virtual roadshow, 3D vlog, medical imaging, and slam.

NeRF uses a group of two - dimensional images with different views as input, generate RGB values and density via MLP network, and uses classic volume rendering to represent synthesized views, which is a three-dimensional scene. In radiance field, NeRF provides volume rendering based on specific input angle of camera. They use the combination of radiance field and volume rendering is because it is differentiable.

To put it simple, the process is to query the corresponding set of colour and density by using spatial location information. we can imagine using NeRF to flatten a space in a specific direction, and the colours in the space are weighted and summed to obtain the colour on the plane. In NeRF, light field is a complete representation of the collection of rays in space. For each ray, multiple points are sampled along its path in the light field. For each sampled point, a 5D coordinate is adopted which include via current point (x, y, z) radiance emits, and towards direction (theta, phi).

A MLP used in NeRF typically consists of several fully connected layers, each followed by a non-linear activation function ReLU (Rectified Linear Unit). The input to the MLP is the 5D coordinate we mentioned above. The output of the MLP is a 4D vector, representing the volume density sigma and RGB value (r, g, b) at that point along that viewing direction. The colour values are then blended using alpha compositing which adopts both the colour and opacity at each point, as well as the distances between the points. The result of this blending process is the colour of a single pixel in the final rendered image.

And then the results are compared with the ground truth, the network parameters are reversely optimized, and continuous iterations are made to make the radiation field close to the true value. The full process please check Fig 1.

A diagram of a network

Description automatically generated

Fig 1: Working Flow of Classical NeRF

This is a departure from traditional computer graphics (explicit geometry), where scenes are typically represented using geometry or raster information, and it allows for a highly compact and efficient representation of complex 3D scenes.

II. 3D Gaussian Representation and Splatting

There are several popular rendering methods for NeRF. Other than the classical volumetric rendering, Signed Distance Function (SDF) is also a very popular way. SDF is a function that gives the shortest distance between a point and a surface. It (the signed function) can differentiate between points inside and outside the surface. The function returns a positive value for points outside the surface (SDF>0), zero on the surface (SDF=0), and a negative value (SDF<0) for points inside the surface. SDF determines that the sampling points closer to the surface have a higher colour contribution.

In NeRF each ray can potentially interact with a different number of objects in the scene, leading to a variable amount of computations needed per ray. And these interactions could possibly cause ray reflection and refraction. It means that computations cannot be done in parallel, which reduces the efficiency of the GPU. Also classical NeRF is run in Pytorch + CUDA environment which mostly are considered run under general-purpose computing on graphics processing units (GPGPU).

While processing traditional rasterization as we all know is highly fast but it is a discrete process, as well as SDF. One of the main contents of traditional rasterization to process 3D scenes is to map three-dimensional meshes to the projection plane and do the pixelation. The projection here is called splatting.

In 3D Gaussian representation, a single 3D Gaussian splatting can be optimized as a small differentiable space, and multiple Gaussians can be rendered in parallel rasterization. This can be seen as a delicate balance between differentiability and discreteness.

But what is 3D Gaussian? 1D Gaussian, as we know, is the normal distribution and the shape is a simple bell symmetric curve as shown in Fig 2(a).

|  |  |  |
| --- | --- | --- |
| A diagram of a normal distribution with Ryugyong Hotel in the background  Description automatically generated | A graph of a function  Description automatically generated | A graph of a sphere  Description automatically generated |

Fig 2(a) Normal distribution: 1D Gaussian from <https://en.wikipedia.org/wiki/Normal_distribution>. 2(b) Visualisation of a 2D Gaussian model from <https://www.hindawi.com/journals/ddns/2006/063245/>. 2(c) Visualization of a 3D Gaussian model from <https://www.mdpi.com/1424-8220/12/7/8640>.

A set of miu and sigma can determine a 1D Gaussian distribution function, and then determine a 1D line segment [miu-3sigma, miu+3sigma]. By changing miu and sigma can express a line segment on the 1D number axis. Similarly, if it is expanded to a three-dimensional Gaussian distribution it is an ellipsoid, as shown in Fig 2(c).

The basic elements of the 3D Gaussian construction can cover enough diverse geometries, as defined below.

The 3D Gaussian and ellipsoid are isomorphic in geometry, and the ellipsoid can be obtained by scaling and rotating the ball along the axis. So we can also do the same to generate a 3D Gaussian as below.

If some 3D Gaussian of a point generated is too large to fully fit the details of that point, the point is segmented and expressed by two points; if the points at a certain point are too dense, multiple 3D Gaussians are merged into one 3D Gaussian, as shown in Fig 3.

A diagram of a clone

Description automatically generated with medium confidence

Fig 3: adaptively adjust point cloud distribution according to gradient.

The axial integral of 3D Gaussian is equivalent to that of 2D Gaussian, which mathematically gets rid of the limitation of sampling amount (unlike the discrete accumulation to fit continuous Integral in classical NeRF).

The amount of calculation is determined by the number of Gaussian, and Gaussian can be quickly rendered in parallel using the rasterization pipeline. Though in the paper they did not provide details of the way to transfer 2D Gaussian to pixel, it is basically a rasterization process to transfer plane triangle is transformed into pixels through algorithms such as scan lines and barycentric coordinate methods. The 3D Gaussian splatting requires a transferring of images to 3D structure of motion (SfM) points as prerequisite, which estimates the 3D structure of a scene from a set of 2D images.

III. Synthesizing Video Streaming to Another

The project of synthesizing one source of video frames to another in real time based on 3D Gaussian splatting contains the following steps:

1. 3D Reconstruction: The first step in this process would be to reconstruct the 3D geometry of the scene from the input video with SfM.

2. SfM Point Cloud Generation: Once the 3D structure of the scene is obtained, a point cloud representation of the scene can be generated. Each point in the point cloud represents a point in the 3D scene, and include additional data such as colour information.

3. 3D Gaussian Splatting: The point cloud data is then processed using 3D Gaussian splatting. Each point is splatted into a Gaussian distribution that adds smoothly to its neighbours, resulting in a smooth 3D field that represents the original point cloud data.

4. Rasterization: The 3D field is then rasterized to produce a 2D image. This involves casting rays from the camera position through each pixel in the 2D image, and sampling the 3D field along each ray to determine the pixel's colour.

5. Video Synthesis: The rasterization process is repeated for each frame in the output video, possibly with a different camera position or other parameters for each frame. This generates a sequence of 2D images that form the output video.

6. Output Streaming: The synthesized video is then streamed to the recipient. This could involve encoding the video in a suitable format, transmitting it over a network, and decoding it on the recipient's end.